Trim Time Reduction in Analog/RF ICs Based on Inter-Trim Correlation

V. A. Niranjan*, D. Neethirajan*, C. Xanthopoulos†, E. De La Rosa‡, C. Alleyne‡, S. Mier‡ and Y. Makris*
*The University of Texas at Dallas, 800 W. Campbell Rd., Richardson, TX, 75080, USA
†Advantest America, 12301 Riata Trace Pkwy, Austin, TX 78727, USA
‡Qualcomm Technologies Inc., 5775 Morehouse Dr., San Diego, CA 92121, USA

2021 IEEE 39th VLSI Test Symposium (VTS) | 978-1-6654-1949-9/20/$31.00 ©2021 IEEE | DOI: 10.1109/VTS50974.2021.9441034

***Abstract*—Post-fabrication performance calibration, a.k.a. trimming, is an essential part of analog/RF IC manufacturing and testing. Its objective is to counteract the impact of process variations by individually fine-tuning the performance parameters of every fabricated chip so that they meet the design specifications and, thereby, to ensure both high yield and high performance. The prevalent trimming process currently employed in industry involves a search algorithm which consists of repeated digital trim-code selection and measurement in order to optimize the trimmed performance. With hundreds of trims commonly performed on contemporary analog/RF chips, this process becomes overly expensive. In this work, we discuss a machine learning-based approach that ameliorates this problem by leveraging inter-trim correlation. Specifically, our method relies on effectively trained regression models which use the measurements obtained through an intelligently selected and conventionally performed subset of trims, in order to accurately predict the optimal trim codes for the omitted trims. Thereby, as corroborated using data from an actual analog/RF IC currently in production, trim time can be drastically reduced without significantly affecting the accuracy of the selected trim codes.**

I. Introduction

The continuous scaling of minimum feature sizes in contemporary semiconductor manufacturing technologies, which seeks to enable more compact yet more powerful integrated circuits (ICs), has introduced increasingly profound process variation and, by extension, new challenges in the IC design, fabrication and test process. Analog/RF ICs, in particular, have been faced with the challenge of a wider and less controlled distribution of fabricated chip performances due to process variation which, in turn, increases the likelihood of chip performances falling outside their design specification range and resulting in excessive yield loss. As a countermeasure, analog/RF designers have traditionally resorted to conservative margins in order to mitigate the risk, thereby leaving performance on the table. Recently, however, in an effort to achieve high performance through aggressive design while, at the same time, ensuring high yield despite the increased process variation of the latest technology nodes, the concept of post-fabrication performance calibration, or *trimming*, has been extensively explored [1]–[15].

Trimming methods rely on "tuning knobs," i.e., circuitry which can be introduced in the design and individually tuned for each fabricated chip after manufacturing, in order to counteract the impact of process variation and bring the performances of the chip not only within the design specifications but also as close to their optimal value as possible. Once the appropriate setting for each such tuning knob is decided, it is saved on-chip, usually as a digital code stored in non-volatile or one-time programmable memory. This trimming process is typically performed during manufacturing testing and results in each chip having its own set of trim codes that optimize its performances. In this way, high yield and high performance can be simultaneously achieved.

The actual trimming process, however, is a rather challenging and time-consuming. Indeed, the relation between knob positions (i.e., trim codes), process variation and analog/RF IC performances is quite complex. Therefore, industrially deployed and practiced solutions involve worst-case exhaustive algorithms which carry out a directed (usually linear or binary) search in the space of possible trim codes, measuring the target performance for each selected trim code and terminating the search when a target criterion (i.e., performance value) has been reached or when there are no more code options to try. Considering the realities of analog/RF IC testing, which involve expensive instrumentation and lengthy setup and settle times, the duration and cost of such an iterative trim code search process becomes quickly quite onerous. Solutions which can reduce the burden of trim code selection without significantly impacting the quality of the chosen trim codes are, therefore, highly desired.

To this end, a variety of approaches have sought to leverage the power of statistical and machine learning methods, along with the correlation amongst continuous measurements in analog/RF ICs, in order to accelerate the process of post-production performance trimming. Specifically, based on the general concepts of *alternate test* [16] and *machine learning-based test* [17], numerous statistical trimming methods have been proposed in the literature, using simple measurements from either intrusive or non-intrusive on-chip sensors. Broadly, these methods can be divided into *iterative test-&-tune* [8], [11], [14] and *one-shot* [1], [9], [13], [18] approaches. Iterative test-&-tune solutions typically perform a directed search, each time selecting a new trim code and repeating the low-cost measurements through which they predict (using trained statistical models) the performances that are being trimmed, ultimately seeking to optimize a target criterion in the space of device performances. One-shot solutions, on the other hand, share the same objective yet use only one set of low-cost measurements for a single trim code in order to build statistical models which enable prediction of the final trim code for a device.

Along a different direction to trim cost reduction, the method in [19] uses a lookup table to estimate the trim code, followed by a linear search in the vicinity of the estimated trim code to arrive at the optimal trim code. In [20], the trim

978-1-6654-1949-9/21/$31.00 ©2021 IEEE

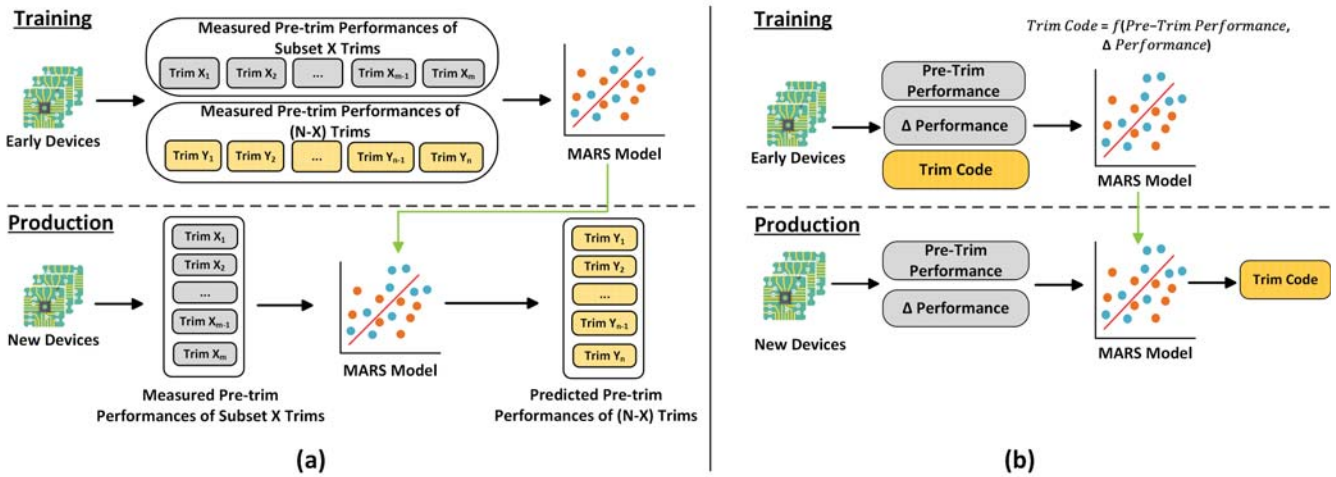

Fig. 1: Training and Application of (a) Pre-Trim Performance Prediction Models, and (b) Trim Code Prediction Models

code is based on a trim code lookup table that is updated dynamically when process shifts in wafer fabrication occur. The method in [21] uses the unique electrical test (E-test) signature of each wafer to customize the starting point and, thereby, narrow the range of the trim code search. In yet another approach to speed up trimming, the method in [22] takes advantage of spatial correlation of trims across a wafer and uses measurements from a few sampled die to predict the trim codes for the remaining die on the wafer. Alternatively, the method in [23] reduces trim time in the case of multi-variable trims by simultaneously searching for the optimal search codes through a simplex algorithm.

In this work, we investigate a different statistical approach to reducing the time and cost of analog/RF IC trimming. Specifically, in contrast to the aforementioned methods which leverage correlation between low-cost (alternate) measurements and circuit performances to select the appropriate trim-code, or which rely on spatial correlation, e-test signatures or simultaneously performed trim correlation, we seek to leverage *inter-trim correlation*. Specifically, our approach relies on measurements obtained while executing a judiciously selected subset of trims using the conventional approach, which are then processed by trained statistical models, in order to predict the trim codes for the remaining trims. In this sense, this method is very compatible and can be easily deployed on the test floor, as it does not require any additional circuitry or measurements beyond what is already available on the chip and the ATE. In a way, our inter-trim correlation-based trim-time reduction solution is similar to statistical analog test compaction [24], [25], which has been extensively researched in the past and wherein a subset of tests are omitted and the corresponding pass/fail decision is predicted as a function of the remaining tests, using trained statistical models. In our case, however, much more elaborate statistical models are required in order to select the trim-codes that optimize the performances of each chip.

II. Proposed Methodology

To effectively reduce the trim time and cost, the proposed methodology sets out to:

- Model the trim codes of a subset of trims as a function of the measurements collected from conventionally performing the rest of the trims.
- Identify appropriately sized subsets of trims which are cost-effective in predicting the remaining trim codes.

Below, we first describe the modeling process and we then investigate two solutions to the optimal trim selection question.

A. Trim Code Modeling

The input to the proposed solution is limited to the measurements obtained while performing a subset of trims on a fabricated chip and, more specifically, the pre-trim performances. The desired output is the integer representation of the optimal trim code for the rest of the trims. Training a model capable of directly predicting the trim codes for a subset of trims based on the pre-trim performance measurements of another subset is challenging, due to the complexity of the underlying relations. To address this challenge, the proposed modeling approach is split into two stages. The first stage models aim to infer the pre-trim performances of the omitted trims based on the pre-trim performances of the conventionally executed trims. In essence, these models serve as feature generators to assist the second stage models and are based on the well-documented existence of performance correlation in analog/RF ICs [16], [17], [24], [25]. The second stage models are, in turn, used to infer the trim code as a function of the pre-trim performance measurement of a chip and the desired post-trim performance value to which the chip must be calibrated.

1) Modeling Stage A - Pre-Trim Performance Prediction: An overview of the first-stage set of models is shown in Figure 1 (a). Prior to deployment of these models, which are based on Multivariate Adaptive Regression Splines (MARS) [26], a training phase is required. Training is performed using a dataset containing the pre-trim performances of both the subset of trims to be retained and the subset of trims to be predicted, measured on a statistically-significant conventionally-trimmed set of chips. During training the MARS model uses its built-in feature selection algorithm to prune features based on a cross-validation score. Herein, we will use N to denote the set of

all trims, X to denote the set of trims which will be retained and executed conventionally in production, and N-X to denote the set of trims that will be predicted. We will also use |N|, |X|, and |N-X| to denote the cardinality of these three sets, respectively. Selection of the subset of |X| trims that should be retained is discussed in Section II-B. During the production phase, the conventional trimming process is performed only for the trims in the X subset. The measured pre-trim performances are, then, passed to the trained models which, in turn, predict the pre-trim performances for the trims in the N-X set.

2) Modeling Stage B - Trim Code Prediction: An overview of the second-stage set of models is shown in Figure 1 (b). Each of these models, which are also based on Multivariate Adaptive Regression Splines (MARS) [26], corresponds to a specific trim in the N-X set. Similar to stage A, a training phase is required, during which the data from a statistically-significant conventionally-trimmed set of devices is used. During training, the pre-trim performance measurement and the Δ performance are used as features, while the trim code is used as the target variable. Δ performance is the difference between the desired post-trim and the pre-trim performance. The trained models can then be used in production to predict the required trim codes based on the pre-trim performance and the desired Δ performance.

Our overall inter-trim correlation-based trim time reduction solution combines the models from the two aforementioned stages. For every produced IC, all trims in set X are performed using the conventional trimming process. The measured pre-trim performances are passed to the corresponding models from Stage A, which predict the pre-trim performances for the N-X subset of trims. These |N-X| predicted pre-trim performances, along with calculated Δ performance required to reach a desired target for each of the |N-X| trims, are passed to Stage B models. Subsequently, the Stage B models infer the trim codes, which are then recorded and stored in the on-die memory to complete the trimming process.

B. Trim Selection Algorithm

Selection of a set of trims with low cardinality and high predictive power is essential to the success of the proposed method. To this end, we explored two trim selection solutions, one based on a statistical heuristic algorithm and the other based on an evolutionary algorithm.

1) Statistical Heuristic Algorithm: The proposed statistical heuristic-based trim selection solution is described in Algorithm 1. As shown, the inputs to the algorithm are the pre-trim performance matrix for all available |N| trims, reciprocal noise sensitivity vector and reciprocal training errors for Stage B models, and the desired number |X| of trims to be selected. The first step of the algorithm calculates the Pearson correlation coefficient matrix for all pre-trim performances. In order to identify a trim that is the best predictor choice, we find the index of the trim that has the maximum sum of absolute correlation coefficients (Algorithm 1, Lines [2-6]). After assigning the identified trim to the X set, an elimination step follows which seeks to exclude trims that are highly correlated to the previously identified trim. This elimination step assists with reducing the redundancy of the selected trims in the X subset. Moreover, the noise sensitivity and training error of Stage B models is taken into account to ensure that the selected trims can be used to build robust trim code prediction models. At the end of this iteration, all identified trims are excluded from further analysis, and the process is repeated until the desired number of trims has been reached.

```
1  def HeuristicTrimSelection(ptf, ts, te, k):
       Input:
         ptf: Pre-trim performances matrix for all N trims,
         ts: Vector of noise sensitivity reciprocals of
       Stage B models for all N trims,
         te: Vector of training error reciprocals of Stage
       B models for all N trims,
         k: number of desired predictors
       Result: List of optimal subset of trims to be used
               as predictors
2      corr_matrix = PairWiseCorrelations(ptf);
3      row_sum = RowSum(corr_matrix);
4      X = [ ];
5      for i in range(k) :
6          max_sum = findMaxIndex(row_sum);
7          X.append(max_sum);
8          while length(NX) < length(ptf)-k :
9              error = [ ];
10             for j in range(length(ptf) :
11                 corr = corr_matrix[max_sum][j];
12                 error.append(corr*ts[j]*te[j]);
13             rhc = findMaxIndex(error);
14             corr_matrix.remove(rhc);
15         corr_matrix.remove(max_sum_index);
16     return X
```

Algorithm 1: Statistical Heuristic-Based Trim Selection

2) Genetic Algorithm: Genetic algorithms are well suited for optimization problems with a large search space, especially when the cardinality of the solution (i.e., the desired number of retained trims) is undefined. Herein, we use NSGA-II [27], an elitist multi-objective genetic algorithm, characterized by the following evolutionary parameters and operators:

- **Initial population:** The solution domain is represented as a gene, consisting of an array of ones representing the trims belonging to the subset X and zeros representing the trims belonging to the subset N-X. The initial gene pool is randomly generated to increase diversity.
- **Fitness function:** Two fitness functions drive the gene selection of the Genetic Algorithm, one penalizing genes with a higher number of 1s (i.e., trims assigned to subset X) and the other penalizing higher compounded prediction errors for Stage A and Stage B models. The compounded prediction error fitness function is defined as:

$$error\_1 = \frac{1}{k}\sum_{j}^{k}\left\{\frac{(pt\_act_j - pt\_pred_j)^2}{USL_j - LSL_j}\right\} \quad (1)$$

$$error\_2 = \frac{1}{k}\sum_{j}^{k} |tc\_act_j - tc\_pred_j| \quad (2)$$

$$fitness\_function\_2 = \prod^{|N|-|X|} \{error\_1 \times error\_2\} \quad (3)$$

where, pt_act and pt_pred are the actual and predicted pre-trim performances, USL_j and LSL_j are the upper and lower specification limits for the j^{th}-trim performance, tc_act and tc_pred are the actual and predicted trim codes, k is the number of devices available, and $|N|$, $|X|$ represent the cardinality of set N and subset X, respectively.

III. Experimental Results

To evaluate the proposed trim time reduction methodology, we experimented with a dataset comprising measurements from fabricated and conventionally trimmed devices. Herein, we first provide details describing the dataset and then present the flow of our experiment and discuss our findings.

A. Dataset

Our dataset comes from an RF transceiver IC currently in production in an advanced technology node and contains measurements obtained on 5,712 devices. Each device is subjected to a set N of trims, with |N|=173, involving a worst-case exhaustive search algorithm in the space of trim-codes, whereby the optimal trim code is identified for every trim on every device. For each of the 173 trims and for every one of the 5,712 devices, our dataset contains the pre-trim performance, the default trim code for which it was measured, the post-trim performance, the final trim code for which this was achieved, as well as the upper and lower specification limits. Without loss of generality, we assume that each trim search incurs, on average, the same cost. Therefore, time savings from predicting trim codes instead of executing the conventional trim searches is linearly proportional to the number of predicted trims. A weighted version of the problem may also be formulated as a straightforward extension of this work to account for differences in trim code search cost.

B. Experimental Flow & Results

Our experiment starts with randomly dividing the dataset into a training set and a test set of equal size (i.e., 50/50 split). The training set is used for the purpose of training the two sets of regression models, as described in Section II, as well as for selecting the subset of X of trims that are conventionally performed through the heuristic of Section II-B or the genetic algorithm of Section II-B2. The test set, on the other hand, is used for evaluating the accuracy of the trained models and the trim selection solutions. Using these two datasets, we seek to evaluate the accuracy of the following:

Pre-Trim Performance Prediction Models: First, we sweep the number of retained trims from |X|=120 down to |X|=40 in increments of 20. For each value of |X|, we select randomly a subset of |X| trims and we use the training set to train regression models for predicting the remaining |N|-|X| pre-trim performances from the retained |X| pre-trim performances. The trained models are then applied to the test set to obtain the predicted pre-trim performances and compare them with the measured ones. The box-plots in Figure 2 summarize the prediction error for each value of |X|. For each device in the test set and for each predicted trim, the Percentile Specification Error (PSE) is computed as the absolute difference of the predicted and actual pre-trim performance, expressed as a percentage of the specification range for this performance (i.e., upper limit - lower limit). The results reveal that:

(i) Very high correlation exists between pre-trim performance measurements. Indeed, even when randomly choosing only 40 out of the 173 pre-trim performances as predictors, we can still predict the remaining 133 with an average PSE of approximately 12%.

(ii) PSE decreases as |X| increases. Indeed, the above PSE number drops to approximately 9% for |X|=80 and 7% for |X|=120, as more correlation can be exploited when more trims are performed. We note that this result is hardly surprising, as the high correlation between analog/RF IC performances has been extensively studied and documented in the literature (e.g., alternate test [16], machine learning-based test [17], analog test compaction [24], [25]).

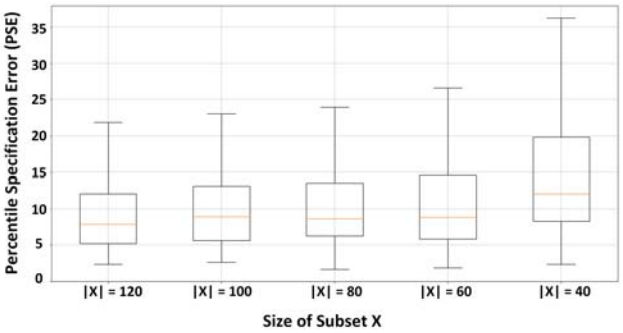

Fig. 2: Accuracy of Pre-Trim Performance Prediction Models

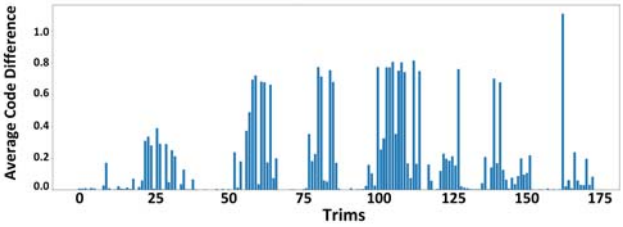

Fig. 3: Accuracy for Trim Code Prediction Models

Trim-Code Prediction Models: Next, we use the training set in order to train a regression model for each of the 173 trims. These models use the measured pre-trim performance along with the difference between the target post-trim performance and the measured pre-trim performance in order to predict the trim-code required to achieve the target. These models are then applied to the test set and the results are summarized in Figure 3, where the average difference between the predicted and actual trim code is provided for each of the 173 trims. The results confirm that the accuracy of these models is very high: on average, the difference between the actual trim code and the predicted trim code is a small fraction of one code value away. Considering that these trim codes can take one of

either 32 or 64 values, this error is almost negligible.

Complete Method: Random vs. Heuristic Trim Selection: Having confirmed the accuracy of the two sets of constituent models, we proceed to evaluating the overall trim time reduction method proposed in Section II. Specifically, we sweep the number of retained trims from |X|=120 down to |X|=40 in increments of 20 and, for each value of |X|, we select either randomly or through the proposed heuristic a subset of |X| trims. Using the training set, we train regression models for predicting the remaining |N|-|X| pre-trim performances from the retained X pre-trim performances. The trained models are then applied to the test set, in order to predict the|N|-|X| pre-trim performances from the selected |X| pre-trim performances. Next, the |N|-|X| predicted pre-trim performances for the devices in the test set are processed by the the trim-code prediction models, which were trained as described in the previous paragraph, to predict the trim code for the target performance. As a target performance, we use the post-trim performance that each device was trimmed to using the conventional approach, since for that value we know the actual trim-code and can, therefore, use it as ground truth to quantify the accuracy of our method. The results for the 5 selected values of |X|, for both random and heuristic selection of the |X| trims are summarized and contrasted in Figures 4 (a)-(e).

In each of these 5 plots, the linear-scale x-axis represents the absolute difference between the actual and the predicted trim code, while the logarithmic-scale y-axis represents the number of predictions that were made. This number is equal to the number of devices in the test set (i.e., 2,856) multiplied by the number of trim codes that were predicted (i.e., 173-|X|). Results for random and heuristic selection are shown side-by-side for ease of comparison. In essence, for each value of trim-code error, the corresponding pair of blue and orange bars reflects how many trim-code predictions were made with such error by our method, when selection of the |X| trims is done through heuristic and random approach, respectively. Based on these results, we point out the following:

(i) Overall, the proposed method is very effective in predicting the trim code required to meet a post-trim performance target. Indeed, for the vast majority of predictions, our method yields the same trim code as the conventional trim code search. For example, as shown in Figure 4 (a), among the 151,368 trims that were predicted for our test set when |X|=120, 148,492 (i.e., 98.10%) were predicted with zero error for heuristic selection and 127,543 (i.e., 84.26%) for random selection. Furthermore, even when the prediction error is not zero, the absolute difference between the actual and predicted trim code is very small. For example, only a handful of trim codes were predicted with an error of more than 3 codes away when |X|=120 and heuristic selection is used.

(ii) As the number of conventionally performed trim searches decreases, the accuracy of the proposed method gradually also degrades. This is expected, as a smaller value of |X| offers fewer measurements and, thereby, less correlation that can be leveraged to predict the remaining trim-codes. This degradation, however is very graceful. For example, as shown

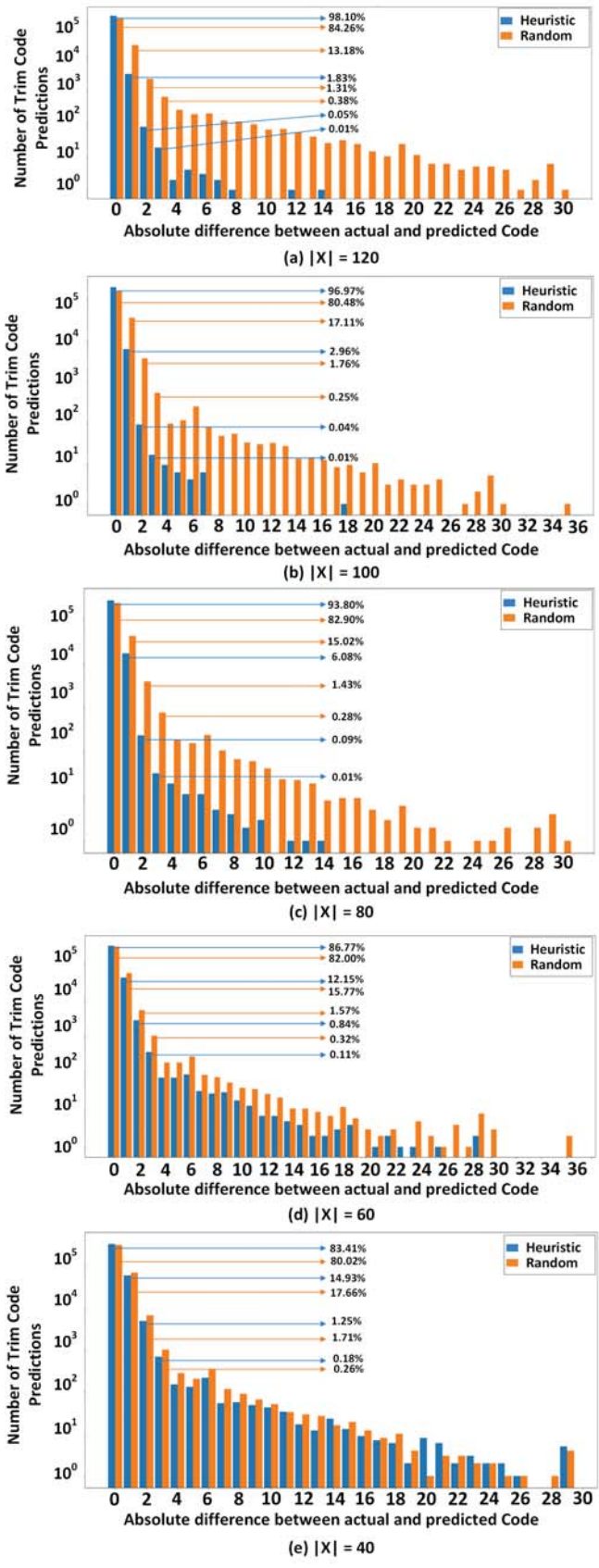

Fig. 4: Accuracy of Proposed Trim Code Prediction Method

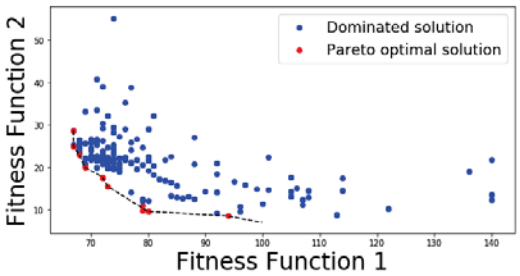

Fig. 5: NSGA-II Pareto Front Evaluation

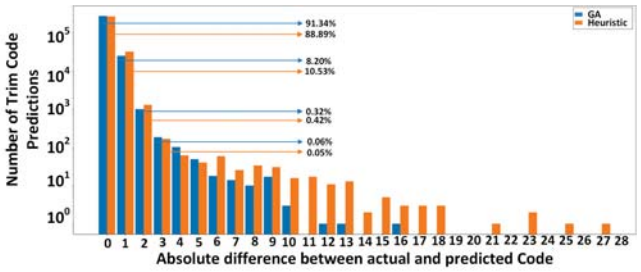

Fig. 6: GA vs. Heuristic Comparison for |X| = 66

in Figure 4 (c), among the 265,608 trims that were predicted for our test set when |X|=80, 249,149 (i.e., 93.80%) were predicted with zero error for heuristic selection and 220,189 (i.e., 82.90%) for random selection. Similarly, as shown in Figure 4 (e), even when only X=40 trims are conventionally performed, among the 379,848 trims that were predicted for our test set, 316,838 (i.e., 83.41%) were predicted with zero error for heuristic selection and 303,954 (i.e., 80.02%) for random selection. It is also important to note that our method enables exploration of the trade-off between the trim code prediction error and the savings of the proposed method, as expressed through lowering the value of |X|.

(iii) The proposed greedy heuristic significantly outperforms the random selection approach. Not only does it result in a slightly higher percentage of trim codes predicted with zero error across all five charts in Figure 4, but also it drastically reduces the difference between the actual and the predicted codes when the error is non-zero. This is reflected in the fact that the distributions of blue bars in Figures 4 (a)-(e) are condensed closer to the y-axis.

Genetic Algorithm-Based Trim Selection: Lastly, we evaluate the effectiveness of the Genetic Algorithm in selecting the subset of |X| trims that should be conventionally performed in order to accurately predict the trim codes for the remaining |N|-|X| trims, as discussed in Section II-B2. Once again, the training set is used for the purpose of evaluating the fitness of the selected subset of conventionally performed trims as the Genetic Algorithm proceeds through multiple generations, while the test set is used for reporting the accuracy of the selected subsets in predicting the remaining trim codes. In this experiment, we ran the NSGA-II algorithm with an initial population of 10 randomly populated genes and we allowed it to proceed through 60 generations. Fitness of the surviving genes from every generation are plotted in the 2-dimensional fitness space in Figure 5, depicting the pareto-optimal front that has been explored by the Genetic Algorithm. The pareto-optimal front, which includes the solutions shown in red, consists of the best choices found by the algorithm. Solutions show in blue, on the other hand, are dominated by the pareto-front and should not be selected as better options exist.

To further assess the quality of the pareto-optimal solutions, we compare the accuracy of the complete trim code selection method when the subset of |X| conventionally performed trims is chosen through the Genetic algorithm and the heuristic approach, respectively. Figure 6 summarizes the results of this comparison for a randomly chosen value of |X|=66. Once again, the linear-scale x-axis represents the absolute difference between the actual and the predicted trim code, while the logarithmic-scale y-axis represents the number of predictions that were made, in this case 2,856 devices multiplied by 107 predicted trims, for a total of 305,592 predictions. For each value of trim-code error, the corresponding pair of blue and orange bars reflects how many trim-code predictions were made with such error by our method, when selection of the |X| trims is done through the genetic algorithm and the heuristic approach, respectively. The results show that, when NSGA-II is used to select the |X| trims that are conventionally performed, 279,128 out of the 305,592 trim code predictions (91.34%) have zero error, while when the heuristic approach is used, this number is somewhat smaller, namely 271,640 (88.89%). Furthermore, the blue distribution is condensed closer to the y-axis, revealing that even through a modest number of 60 generations, the genetic algorithm results in smaller error than the heuristic approach for predictions where the error is not zero. Similar results following the same trends have been observed for all other points of the pareto-optimal font, confirming that the Genetic Algorithm is superior to the heuristic approach for selecting the subset of |X| trims that should be conventionally performed.

IV. Conclusion

This work investigated the effectiveness of a machine learning-based approach to reducing the time required for trimming the performances of a fabricated analog/RF IC by leveraging correlation across multiple trims. The proposed approach selects a subset of trims to be performed in the conventional manner and utilizes the pre-trim performance measurements of these trims to predict the pre-trim performance measurements of the remaining trims through trained non-linear regression models. Subsequently, it uses these predicted pre-trim performances in order to predict the trim code that is required to achieve a post-trim performance target through a second set of trained non-linear regression models. As demonstrated experimentally using actual data from a contemporary analog/RF IC currently in production, intelligent selection of the subset of trims that need to be conventionally performed, alongside efficient training of the regression models, facilitates accurate prediction of the optimal codes for the omitted trims while drastically reducing the number of retained trims and, thereby, the time and cost of executing the corresponding trim-code search algorithms.

REFERENCES

[1] M. Andraud, H. G. Stratigopoulos, and E. Simeu, "One-shot non-intrusive calibration against process variations for analog/RF circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 11, pp. 2022–2035, 2016.

[2] G. Volanis, D. Maliuk, Y. Lu, K. S. Subramani, A. Antonopoulos, and Y. Makris, "On-die learning-based self-calibration of analog/RF ICs," in *IEEE VLSI Test Symposium (VTS)*, 2016, pp. 1–6.

[3] T. Zhu, M. B. Steer, and P. D. Franzon, "Surrogate model-based self-calibrated design for process and temperature compensation in analog/RF circuits," *IEEE Design Test of Computers*, vol. 29, no. 6, pp. 74–83, 2012.

[4] D. Chang, J. Kitchen, and S. Ozev, "Post-production adaptation of RF circuits for application-specific performance metrics," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2016, pp. 2775–2778.

[5] M. El-Nozahi, E. Sanchez-Sinencio, and K. Entesari, "A CMOS low-noise amplifier with reconfigurable input matching network," *IEEE Transactions on Microwave Theory and Techniques*, vol. 57, no. 5, pp. 1054–1062, 2009.

[6] K. Jayaraman, Q. Khan, B. Chi, W. Beattie, Z. Wang, and P. Chiang, "A self-healing 2.4GHz LNA with on-chip S11/S21 measurement/calibration for in-situ PVT compensation," in *IEEE Radio Frequency Integrated Circuits Symposium (RFIC)*, 2010, pp. 311–314.

[7] T. Das, A. Gopalan, C. Washburn, and P. R. Mukund, "Self-calibration of input-match in RF front-end circuitry," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 52, no. 12, pp. 821–825, 2005.

[8] V. Natarajan, S. Sen, A. Banerjee, A. Chatterjee, G. Srinivasan, and F. Taenzler, "Analog signature-driven postmanufacture multidimensional tuning of RF systems," *IEEE Design Test of Computers*, vol. 27, no. 6, pp. 6–17, 2010.

[9] D. Han, B. S. Kim, and A. Chatterjee, "DSP-Driven self-tuning of RF circuits for process-induced performance variability," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 2, pp. 305–314, 2010.

[10] S. M. Bowers, K. Sengupta, K. Dasgupta, B. D. Parker, and A. Hajimiri, "Integrated self-healing for mm-wave power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 3, pp. 1301–1315, 2013.

[11] V. Natarajan, A. Banerjee, S. Sen, S. Devarakond, and A. Chatterjee, "Yield recovery of RF transceiver systems using iterative tuning-driven power-conscious performance optimization," *IEEE Design Test*, vol. 32, no. 1, pp. 61–69, 2015.

[12] S. Larguech, F. Azaïs, S. Bernard, M. Comte, V. Kerzerho, and M. Renovell, "Efficiency evaluation of analog/RF alternate test: Comparative study of indirect measurement selection strategies," *Microelectronics Journal*, vol. 46, no. 11, pp. 1091 – 1102, 2015.

[13] Y. Lu, K. S. Subramani, H. Huang, N. Kupp, K. Huang, and Y. Makris, "A comparative study of one-shot statistical calibration methods for analog/RF ICs," in *IEEE International Test Conference (ITC)*, 2015, pp. 1–10.

[14] S. Sun, F. Wang, S. Yaldiz, X. Li, L. Pileggi, A. Natarajan, M. Ferriss, J. O. Plouchart, B. Sadhu, B. Parker, A. Valdes-Garcia, M. A. T. Sanduleanu, J. Tierno, and D. Friedman, "Indirect performance sensing for on-chip self-healing of analog and RF circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 8, pp. 2243–2252, 2014.

[15] G. Leger and M. J. Barragan, "Brownian distance correlation-directed search: A fast feature selection technique for alternate test," *Integration*, vol. 55, pp. 401–414, 2016.

[16] P. N. Variyam, S. Cherubal, and A. Chatterjee, "Prediction of analog performance parameters using fast transient testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 3, pp. 349–361, 2002.

[17] H.-G. D. Stratigopoulos and Y. Makris, "Error moderation in low-cost machine learning-based analog/RF testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 2, pp. 339–351, 2008.

[18] N. Kupp, H. Huang, P. Drineas, and Y. Makris, "Post-production performance calibration in analog/RF devices," in *IEEE International Test Conference*, 2010, pp. 245–254.

[19] R. Mittal, M. Kawoosa, and R. A. Parekhji, "Systematic approach for trim test time optimization: Case study on a multi-core RF SOC," in *2014 International Test Conference*, 2014.

[20] R. F. Bullag, R. C. Ortega, and S. B. Bullag, "Adaptive trimming test approach — The efficient way on trimming analog trimmed devices at wafer sort," in *36th International Electronics Manufacturing Technology Conference*, 2014.

[21] C. Xanthopoulos, A. Ahmadi, S. Boddikurapati, A. Nahar, B. Orr, and Y. Makris, "Wafer-Level Adaptive Trim Seed Forecasting Based on E-tests," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017.

[22] C. Xanthopoulos, K. Huang, A. Poonawala, A. Nahar, B. Orr, J. M. Carulli, and Y. Makris, "IC laser trimming speed-up through wafer-level spatial correlation modeling," in *2014 International Test Conference*, 2014.

[23] P. Bongale, V. Sundaresan, P. Ghosh, and R. Parekhji, "A novel technique for interdependent trim code optimization," in *2016 IEEE 34th VLSI Test Symposium (VTS)*, 2016.

[24] H.G. Stratigopoulos, P. Drineas, M. Slamani, and Y. Makris, "Rf specification test compaction using learning machines," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 6, 2010.

[25] S. Biswas and R. D. Blanton, "Reducing test execution cost of integrated, heterogeneous systems using continuous test data," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 1, pp. 148–158, 2011.

[26] J. H. Friedman, "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, 1991.

[27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," in *IEEE Transactions on Evolutionary Computation*, 2002.